ANALYSIS

44(201), October 1, 2015

ISSN 2278–5469
EISSN 2278–5450

Discovery

High speed neuron implementation using Vedic mathematics

Yamuna SV[1], Anshika[2], Nidhi Goel[3], Indu S[4]

1. Department of Electronics and Communication Engineering, Delhi Technological University, Delhi, India; Email: yamuna1802@gmail.com
2. Department of Electronics and Communication Engineering, Delhi Technological University, Delhi, India; Email: ansh.delhi89@gmail.com
3. Department of Electronics and Communication Engineering, Delhi Technological University, Delhi, India; Email: nidhi.iitr@gmail.com
4. Department of Electronics and Communication Engineering, Delhi Technological University, Delhi, India; Email: s.indu@rediffmail.com

Article History

Received: 30 July 2015

Accepted: 13 September 2015

Published: 1 October 2015

Citation

Yamuna SV, Anshika, Nidhi Goel, Indu S. High speed neuron implementation using Vedic mathematics. *Discovery*, 2015, 44(201), 25-32

ABSTRACT

Neural network based systems are said to be much faster than their counterparts. Artificial Neural Network has been considered to speed up processing due to its parallel architecture and the numerous mathematical computations involved for processing any input. In the present work, we have explored the possibility of enhancing the computation speed in a single neuron. This is achieved by performing the neural network computations using Vedic mathematics rather than the conventional addition multiplication process. The multiplier is based on the 'Urdhva Tiryagbhyam' sutra, which is one out of sixteen sutras of ancient Vedic mathematics. Here, coding is done in VHDL and synthesis is done in Xilinx ISE series using Modelsim. Several experiments have been conducted to analyze the results. Comparative analysis indicates the improvement in computation speed of neuron designed using Vedic multiplier over the conventional multiplier. This increase in processing speed can be of use in several real time operations where speed is critical.

Keywords: Artificial neural network; Vedic multiplication; Urdhva Tiryagbhyam

1. INTRODUCTION

The artificial neural networks are used for a number of applications including pattern recognition, classification, identification, GPS systems, speech, vision, and control systems [1-4]. They can also be trained to solve things that are difficult for conventional computers or human beings. Additionally, they have attractive properties like adaptiveness, self-organization, nonlinear network processing and parallel processing. This has lead to the use of neural network in applications involving classification, association, decision-making and reasoning [5]. Artificial neural networks consist of massively parallel network and require parallel architecture for high speed operations in real time applications [6].

Multiplier is a crucial block in designing arithmetic, signal and image processors; such high speed multipliers play an important role in designing an efficient architecture [7, 8]. Recently, several high speed multipliers designed using Vedic mathematics have been reported [9, 10]. The use of Vedic mathematics lies in the fact that it reduces the typical calculations in conventional mathematics into very simple one and reduces the computation time. This is because the Vedic formulae are claimed to be based on the natural principles on which the human mind works. Vedic mathematics is a collection of arithmetic rules that allow more efficient speed implementation.

In this paper, features of ANN & Vedic multiplier are used to estimate the performance of a neuron designed with Vedic multiplier against normal multiplier. Also, the hardware implications of both the schemes have been done. Section II presents a brief introduction about ANN and an explanation of Vedic multiplier with Urdhva Tiryagbhyam sutra is given in Section III. Section IV explains about the design and hardware implementation of the proposed work. Section V explains the simulation results of the work. The last section VI deals with the conclusion of the proposed work.

2. ARTIFICIAL NEURAL NETWORKS

A. Introduction to Artificial Neural Networks

The work on artificial network has been motivated from the working of human brain in a way that brain is a highly complex nonlinear and parallel computer with a capability to organize its structural constituents known as neurons so as to perform the certain computation many times faster than that of the digital hardware [4]. The computation can be viewed as a system in which the inputs are received from an external stimulus. The receptors convert the external stimulus into electrical impulses that convey the information to the neural net and depending on the electrical impulses the neural net conveys or transmits the information to effectors to provide an optimal response to stimulus as shown in "Fig.1". The brain is composed of an integral constituent known as neuron. The brain learns to distinguish different patterns by training these neurons which give the appropriate output [11].

In order to emulate the brain, the concept of artificial neural network has been designed to model the way in which the brain performs a particular task or function of interest. A neural network is a massively parallel distributed network made up of simple processing units. These processing units have a natural tendency of storing experiential knowledge and making it available for use [12]. It is analogous to the brain in two aspects:

- The network obtains knowledge from its environment through a learning process.
- Acquired knowledge is stored in interneuron connection strengths known as synaptic weights.

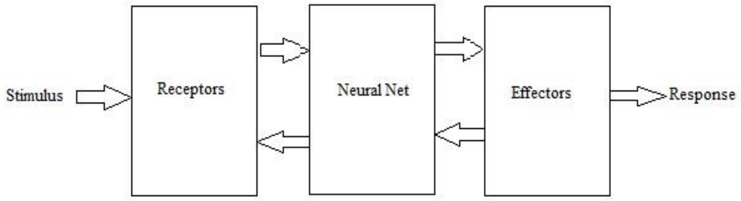

Figure 1 Block Diagram Representation of Nervous system

B. Model of Neuron

In 1958, Rosenblatt introduced the mechanics of the single artificial neuron and introduced the 'Perceptron'. Basic findings from the biological neuron operation enabled early researchers (e.g., McCulloch and Pitts, 1943) to model the operation of simple artificial neurons. An artificial processing neuron receives inputs as stimuli from the environment, combines them in a special way to form a 'net' input, passes that over through a linear threshold gate, and transmits the signal forward to another neuron [13]. If the sum of all the weighted inputs is above the threshold, the output is fired else not. The nonlinear (McCulloch) model of neuron is as shown in "Fig. 2". The neuron may be described in the following way

$$u_k = \sum_{j=1}^{n} w_{kj} x_j \qquad (1)$$

$$y_k = \emptyset(u_k + b_k) \qquad (2)$$

where x_1,x_2,..x_m are the input signals.

w_{kj}refers to synaptic weight of neuron *k*

$y_k$is the output of neuron *k*.

The relationship between induced local fields or the activation potential v_k of the neuron depends on whether bias b_k is positive or negative and is given as follows:

$$v_k = u_k + b_k \qquad (3)$$

$$v_k = \sum_{j=0}^{m} w_{kj} x_j \qquad (4)$$

$$y_k = \emptyset(v_k) \qquad (5)$$

Depending on the application requirement, various activation functions could be used. Usually, three basic type of activation function i.e. threshold function, piecewise linear function and sigmoid function are used.

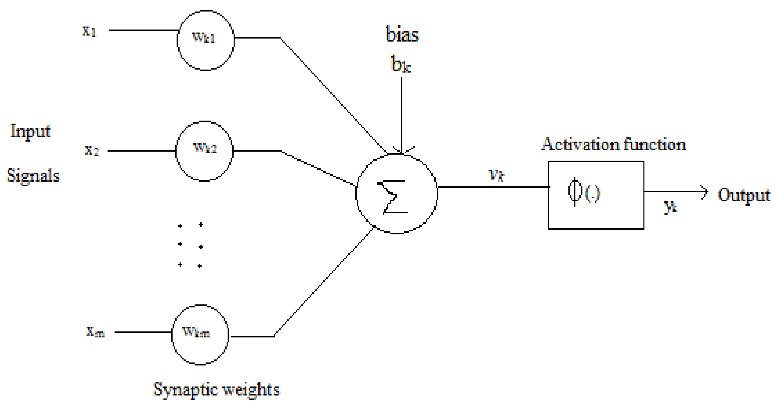

Figure 2 Nonlinear model of Neuron [4]

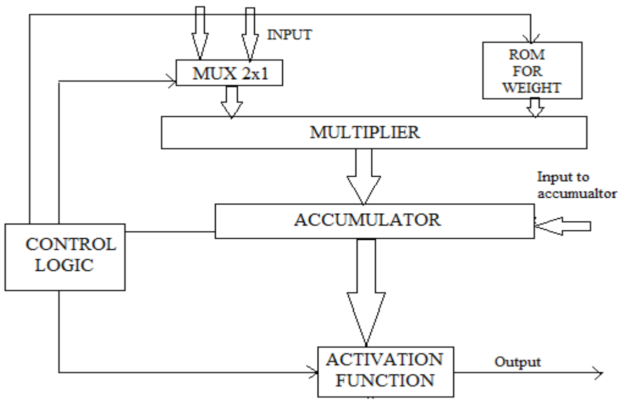

Figure 3 Block Diagram of a Single Neuron

C. Hardware Implementation of a Neuron

Though hardware implementation of the artificial neural network has a long history, over 25 years by now [14], still it is the centre of attention for many researchers [15], while its FPGA implantation remains a challenge [16,17,18].FPGAs are devices which enable the implementation of hardware components in a reconfigurable way. This fact makes that FPGAs are cheaper than ASICs (since FPGAs are reconfigurable so many times, we do not need a new FPGA with each application). Besides the software flexibility, FPGAs present the advantage of the hardware execution speed, with their higher parallelism exploitation. Finally, FPGAs provide a significant improvement in the performance/cost ratio compared with ASICs [19]. The implemented non-linear model of a single neuron is shown in “Fig. 3”.

3. INTRODUCTION TO MULTIPLIER

Multiplication is an essential and basic function in arithmetic procedures and Vedic mathematics is an ancient methodology developed by Indians, which is capable of faster intellectual computation. Multiplication is one of the most important operations in digital neural network based systems. There are several Multiplication-based operations. Frequently used such Computation-Intensive Arithmetic Functions are multiply and Accumulate (MAC) and inner product [20]. Further, multipliers play a major role in the overall power consumption of any system. Therefore, reducing their power dissipation satisfies the overall power budget of ANN system. Higher throughput arithmetic operations are important to achieve the desired performance in many real-time signal and image processing applications [21]. One of the key arithmetic operations in such applications is multiplication and the development of fast multiplier circuit has been a subject of interest over decades. The essential requirements for many applications developed using neural network are reduction of the time delay and power consumption.

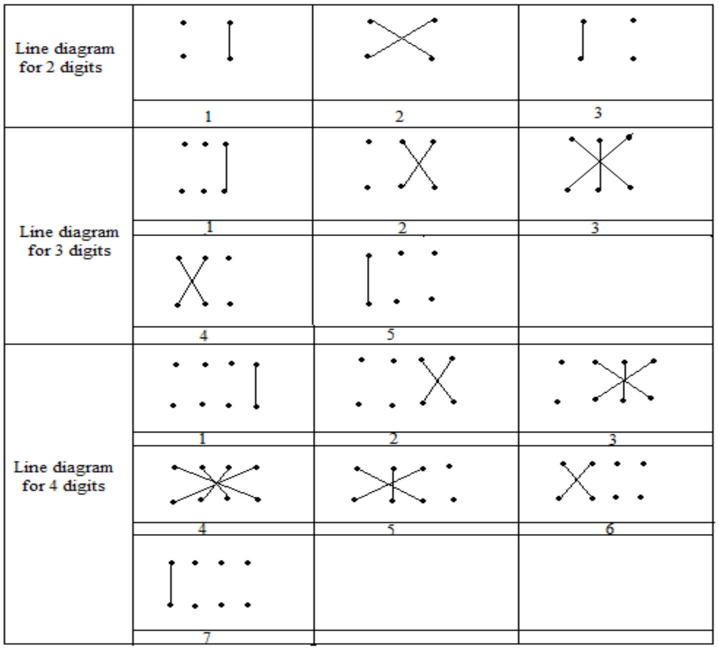
Line diagram for 2 digits
1
2
3
Line diagram for 3 digits
1
2
3
4
5
Line diagram for 4 digits
1
2
3
4
5
6
7

Figure 4 Line Diagram for Vedic Multiplication of 2, 3 and 4 digits

A. Vedic Multiplier

The word “Vedic” is a Sanskrit word derived from the word, ‘Veda’ which means store house of all knowledge. Vedic mathematics is based on 16 Sutras (or aphorisms) that deal with various branches of mathematics like arithmetic, algebra, geometry etc. In Vedic mathematics, methods of basic arithmetic are extremely simple and powerful [22, 23]. In Vedic mathematics, Urdhva tiryagbhyam sutra is used generally for high speed multiplication. Vedic multiplier is very much advantageous in case of bigger multiplication which would consume more time in a conventional multiplier.

B. Details of Urdhva-Tiryagbhyam Method of Multiplication

Urdhva Tiryagbhyam sutra is a sutra for multiplication in the ancient Indian Vedic mathematics. This is based on the generation of all partial products by means of concurrent addition of the partial product. Urdhava Tiryagbhyam obtains parallelism in generation of partial products and their summation. Vedic mathematics reduces the typical calculations in conventional mathematics into very simple one. This sutra uses Vedic formulae that are based on the natural principles on which the human brain works. Vedic mathematics is a technique of arithmetic rules that allow more efficient speed implementation. Mostly, Urdhva Tiryagbhyam is applicable to all cases of multiplication. 'Urdhva' and 'Tiryagbhyam' words are derived from Sanskrit literature in which 'Urdhva' means 'Vertically' and 'Tirya-gbhyam' means 'Crosswise'. Therefore, this algorithm uses the concept of vertical and crosswise multiplication and addition to get the partial products.

C. Algorithm

To illustrate this multiplication scheme, the multiplication of two decimal numbers is chosen. Line diagram for the multiplication is shown in “Fig. 4”.The line diagram demonstrates the procedure for multiplication of 2, 3 and 4 bits. In the line diagram, the dots represent bit “0” or “1”. From the line diagram of 2 bit multiplication, it is seen that the digits on both sides of the line are multiplied and added with the carry from the previous step [24]. The result generates one of the bits of the result and a carry. In the next step, this carry is added and hence the process goes on. The same procedure is followed for more number of bits. The line diagram shows the methodology adapted for 3 and 4 bits [25].

D. Urdhva Multiplier Hardware Architecture

The 'Urdhva Tiryagbhyam' algorithm can be implemented for binary number system in the same way as decimal number system. As an example, a 4x4 Vedic multiplier hardware implementation is explained. The 4x4 bit Vedic multiplier module is implemented using four 2x2 bit Vedic multiplier modules. Let’s examine 4x4 multiplication say a=a3a2a1a0 and b=b3b2b1b0. The output for the multiplication result is s7s6s5s4 s3s2s1s0. Divide a and b into two parts say a3a2 & a1a0 for a and b3b2 & b1b0 for b using the fundamentals of Vedic multiplication, by taking two bits at a time and using 2 bit multiplier block. The structure for multiplication will be as shown in “Fig. 5”. Each block, as shown is a 2x2 bit Vedic multiplier. The inputs to the first multiplier are a1a0 and b1b0. The last block is a 2x2 bit multiplier with inputs a3a2 and b3b2. The middle one shows two 2x2 bit multipliers with inputs a3a2 & b1b0 and a1a0 & b3b2. The final result of multiplication is of 8 bits, s7s6s5s4 s3s2s1s0. To get final product (s7s6s5s4 s3s2s1s0) four 2x2 bit Vedic multiplier and three 4-bit Ripple-Carry (RC) adders are required.

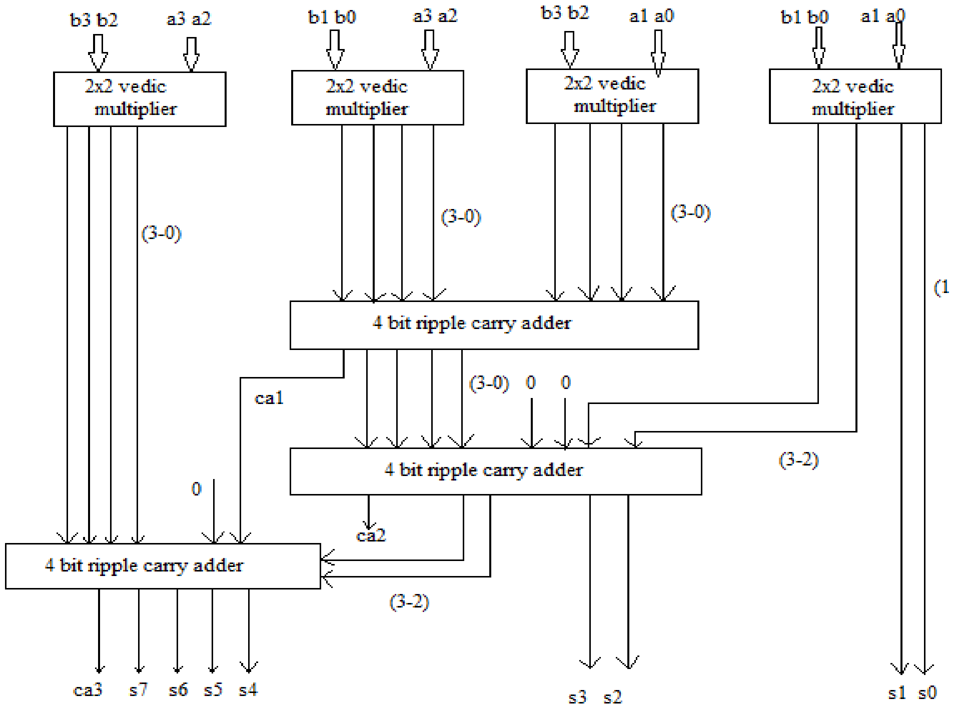

Figure 5 Block Diagram of 4x4 bit Vedic Multiplier

4. PROPOSED HARDWARE ARCHITECTURE OF NEURON IMPLEMENTED USING VEDIC MULTIPLIER

The proposed hardware is almost similar to a standard non-linear neuron (McCulloch Pitts model) except for the multiplier. Here the standard multiplier is replaced with a Vedic multiplier which multiplies the inputs with weights. For evaluation purpose the weights were randomly fixed. “Fig. 3”, shows the proposed architecture of a single neuron which has been designed using Vedic multiplier instead of standard multiplier, to achieve higher speed in designing neural nets.

5. RESULTS

A. Simulation Results

For the purpose of evaluation, performance of a single neuron(with fixed weights) with standard multiplier and Vedic multiplier were compared. The hardware implementation was carried out using VHDL. For implementation, testing and simulation purpose, Xilinx ISE 6.1 with ModelSim simulation tools were used. Spartan II family was been chosen as target device.The simulation and synthesis results are as shown in Fig. 6-7 & Tables 1-3.

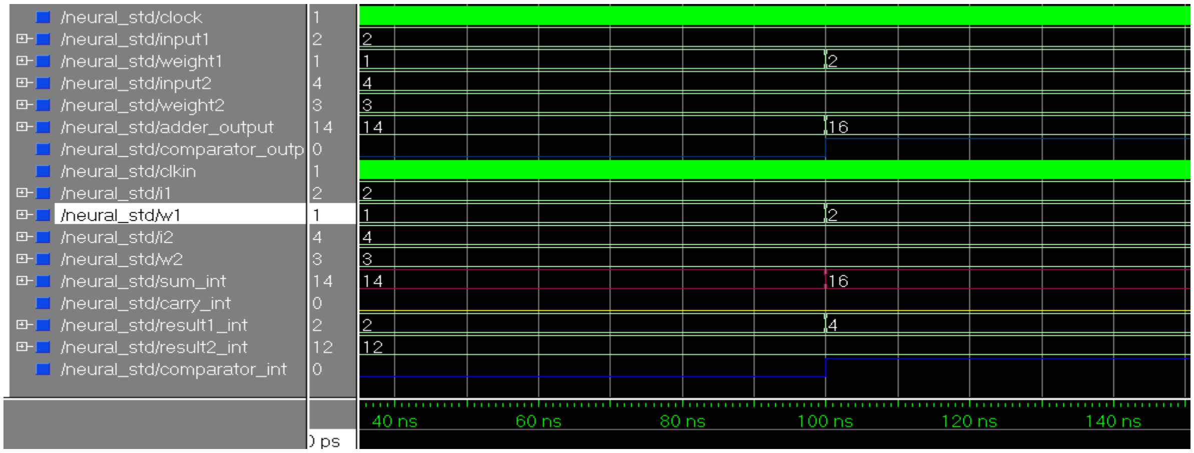

Figure 6 Simulation results of a neuron using standard multiplier

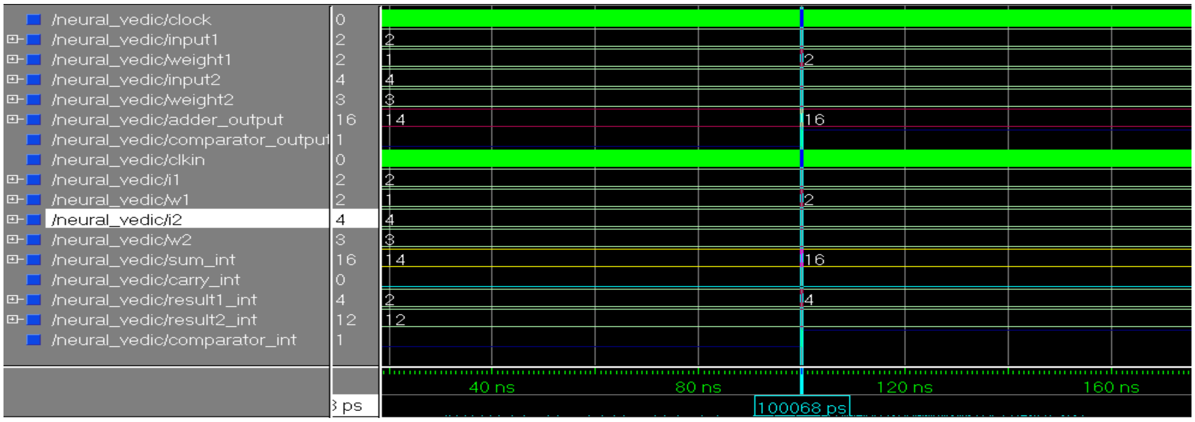

Figure 7 Simulation results of a neuron using Vedic multiplier

Table 1 Synthesis result of neuron with standard multiplier

Logic Utilization	Used	Available	Utilization
Number of Slices	72	192	37%
Number of 4 input LUTs	125	384	32%
Number of bonded IOBs	25	90	27%

Table 2 Synthesis result of neuron with standard multiplier

Logic Utilization	Used	Available	Utilization
Number of Slices	40	192	20%
Number of 4 input LUTs	70	384	18%
Number of bonded IOBs	25	90	27%

Table 3 Delay comparison of neuron

FPGA device package : xc2s15-6-cs144	Delay (ns)	Memory (Kb)
Using standard multiplier	32.969	72772
Using Vedic multiplier	25.675	73668

B. Comparison of Standard Neuron with Vedic Neuron

Table I & II shows the logic utilization of a single neuron realized using standard booth multiplier and Vedic multiplier. From the statistics, it is clear that neuron implemented using Vedic multiplier will utilize less logic compared to the neuron with standard multiplier. Table III compares the result with respect to delay/speed of neural network using standard and Vedic multiplier. The result shows that neuron designed using Vedic multiplier is faster than the one using standard multiplier.

6. CONCLUSIONS

The technologies like ANN & Vedic mathematics (multiplier) are the two important and promising technologies which can increase the processing speed. In this paper, a novel scheme to combine these two technologies to achieve a high performance neuron or ANN has been proposed. A single neuron is implemented using Vedic mathematics (Vedic neuron) in VHDL. The results of standard neuron and Vedic neuron are compared. Hence, the comparison result indicates that the speed of Vedic neuron is better than standard neuron. Also, this research can be extended by replacing Vedic adders in place of standard adders in neurons, which would in all means increase the computation speed. Since, neurons are the basic building blocks of Neurocomputers, the speed of Neurocomputers developed using Vedic neurons would be better than the standard Neurocomputers.

REFERENCE

1. Girish Kumar Jha, “Artificial neural networks and its applications”, Indian Agricultural Research Institute, PUSA, New Delhi,2007.
2. Huqqani A. A, Schikuta E, and Mann E, “Parallelized neural networks as a service in neural networks”, (IJCNN) International Joint Conference on IEEE,2014, pp. 2282-2289 .
3. T. Theocharides, G. Link, N. Vijaykrishnan, M. J. Irwin, and V. Srikantam, “A generic reconfigurable neural network architecture implemented as a network on Chip,” Proc. of the IEEE International System on Chip Conference, September 2004, pp.191-194.
4. Omondi, Amos R., and Jagath Chandana Rajapakse, eds. “FPGA implementations of neural networks”, Vol. 365. New York,USA, Springer, 2006.
5. Janusz Starzyk, and Jing Pang, “Evolvable binary artificial neural network for data classification”, PDPTA - Parallel and Distributed Processing Techniques and Applications, Distributed & Parallel Computing,1995-2013.
6. Chun-Hsien Chen and Vasant Honavar, “A Neural-Network Architecture for Syntax Analysis”, IEEE Transactions on Neural Networks, vol. 10, no. 1,January 1999, pp. 94-114.
7. H.Guilt, "Fully iterative fast array for binary multiplication", Electronics Letters, vol. 5, 1969, pp. 263.

8. Ramalatha, M Dayalan, K D Dharani, P Priya, and S Deborah, “High speed energy efficient ALU design using Vedic multiplication techniques”,ICACTEA, 2009., Jul 15-17, 2009, pp. 600-603.
9. Kunchigi, Vaijyanath, Linganagouda Kulkarni, and Subhash Kulkarni. “High speed and area efficient vedic multiplier”, (ICDCS) International Conference on Devices, Circuits and Systems. IEEE, 2012, pp. 360-364.
10. Y.bansal, C.Madhu, and P.Kaur, “High speed vedic multiplier designs:A review,” Proceedings of IEEE, Recent Advances in Engineering and Computational Sciences (RAECS) 2014, 06-08 March 2014.
11. Haykin, S. “Neural networks a comprehensive foundation”, Prentice Hall Publishing, New Jersey 07458, USA, Vol.1, 1999, 2nd edition, pp 6-7.
12. Bose, N. K, and Liang P, “Neural network fundamentals with graphs, algorithms, and applications”, McGraw-Hill, Inc, Hightstown, NJ, USA, 1996.
13. I.A. Basheer, and M. Hajmeer, “Artificial neural networks: fundamentals, computing, design and application”, Elsevier Science B.V, Journal of Microbiological Methods, vol.43, 2000,pp 3–31.
14. J. Misra,and I. Saha, “Artificial neural networks in hardware: A survey of two decades of progress”, Elsevier, Neurocomputing vol.74, 2010, pp. 239-255.
15. Tisan, M.N. Cirstea, S. Oniga ,and A. Buchman, “ Artificial olfaction system with hardware on-chip learning neural networks”, Proc. of IEEE Int. Conference on Optimisation of Electrical and Electronic Equipment ,2010,pp. 884-889.
16. W. Kurdthongmee, “A novel hardware-oriented Kohonen SOM image compression algorithm and its FPGA implementation”, Journal of Systems Architecture,vol.54, 2008, pp.983–994.
17. Gomperts, A. Ukil, and F. Zurfluh, “Development and implementation of parameterized FPGA-based general purpose neural networks for online applications”, IEEE Transactions on Industrial Informatics, 2011,pp 78-89.
18. Cheng-Jian Lin and Hung-Ming Tsai, “FPGA implementation of a wavelet neural network with particle swarm optimization learning”, Elsevier,Mathematical and Computer Modelling vol. 47,2008, pp 982–996.
19. J. M. Granado, M. A. Vega, R. Pérez, J. M. Sánchez,and J. A. Gómez, “Using FPGAs to Implement Artificial Neural Networks”,13th International Conference on Electronics, Circuits and Systems, ICECS '06, IEEE,2006 , pp 934-937.
20. Purushottam D. Chidgupkar, and Mangesh T. Karad, “The implementation of Vedic algorithms in digital signal processing”, UNESCO International Centre for Engineering Education (UICEE), Australia, vol.8, no.2, 2004.
21. H. Thapliyal, and H.R Arbania. “A Time-Area-Power Efficient Multiplier and Square Architecture Based On Ancient Indian Vedic Mathematics”, Proceedings of the 2004 International Conference on VLSI, Las Vegas, Nevada, June 2004, pp. 434-439.
22. Harpreet Singh Dhillon, and Abhijit Mitra, “A reduced- bit multiplication algorithm for digital arithmetics,” International Journal of Computational and Mathematical Sciences,no.2, 2008.
23. Shamim Akhter, “VHDL implementation of fast NXN multiplier based on Vedic mathematics”, 18th European Conference on Circuit Theory and Design,ECCTD 2007 IEEE, 2007, pp. 472-475.
24. Vinay, “Analysis, Verification and FPGA implementation of Vedic multiplier with BIST capability”,Phd thesis, Thapar University, 2009.
25. Anju, and V.K.Agrawal, “FPGA implementation of low power and high speed Vedic multiplier using Vedic mathematics”. IOSR Journal of VLSI and Signal Processing (IOSR-JVSP), Volume 2, issue 5 (May. – Jun. 2013), pp. 51-57